



(51) 国際特許分類6 G10H 1/00, G11B 20/10, 31/00	A1	(11) 国際公開番号 WO99/40566 (43) 国際公開日 1999年8月12日(12.08.99)
(21) 国際出願番号 PCT/JP99/00557 (22) 国際出願日 1999年2月9日(09.02.99) (30) 優先権データ 特願平10/27584 1998年2月9日(09.02.98) JP (71) 出願人 (米国を除くすべての指定国について) ソニー株式会社(SONY CORPORATION)[JP/JP] 〒141-0001 東京都品川区北品川6丁目7番35号 Tokyo, (JP) (72) 発明者 ; および (75) 発明者/出願人 (米国についてのみ) 辻 実(TSUJI, Minoru)[JP/JP] 今井憲一(IMAI, Kenichi)[JP/JP] 小池 隆(KOIKE, Takashi)[JP/JP] 〒141-0001 東京都品川区北品川6丁目7番35号 ソニー株式会社内 Tokyo, (JP) (74) 代理人 弁理士 小池 晃, 外(KOIKE, Akira et al.) 〒105-0001 東京都港区虎ノ門二丁目6番4号 第11森ビル Tokyo, (JP)		(81) 指定国 JP, US, 欧州特許 (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE) 添付公開書類 国際調査報告書
(54) Title: METHOD AND APPARATUS FOR DIGITAL SIGNAL PROCESSING, METHOD AND APPARATUS FOR GENERATING CONTROL DATA, AND MEDIUM FOR RECORDING PROGRAM (54) 発明の名称 デジタル信号処理方法及び装置、制御データ生成方法及び装置、並びにプログラム記録媒体 (57) Abstract Audio control data extracted from SMF data are supplied to an audio decoder by a data decoder. The audio decoder provides an audio output in response to the received audio control signal. As long as the time required for decoding in the audio decoder is sufficiently short, the playback sounds and the audio output are synchronized since the audio control signal has been decoded from the meta-event of the SMF data. <div style="text-align: right; margin-top: 20px;"> 10 デジタル信号処理装置 </div> <pre> graph LR A[SMFデータ] --> 11[データ復号部] 11 -- C --> 13[MIDI音源] 11 -- D --> 12[オーディオ復号部] B[オーディオ入力] --> 12 12 -- E --> E[オーディオ出力] 13 -- F --> F[MIDI出力] </pre> <div style="margin-top: 10px;"> 10 ... DIGITAL SIGNAL PROCESSOR 11 ... DATA DECODER 12 ... AUDIO DECODER 13 ... MIDI SOUND SOURCE A ... SMF DATA B ... AUDIO INPUT C ... MIDI DATA D ... AUDIO CONTROL DATA E ... AUDIO OUTPUT F ... MIDI OUTPUT </div>		

(57)要約

データ復号部によりSMFデータから取り出されたオーディオ制御データは、オーディオ復号部に供給される。オーディオ復号部は、そのオーディオ制御信号を受け、それにしたがって、オーディオ出力を発音している。このオーディオ復号部での復号時間が十分に短ければ、上記オーディオ制御信号はSMFデータのメタイベントから復号したものであるもので、上記演奏音とオーディオ出力は同期のとれた再生音となる。

PCTに基づいて公開される国際出願のパンフレット第一頁に掲載されたPCT加盟国を同定するために使用されるコード(参考情報)

AE	アラブ首長国連邦	ES	スペイン	LI	リヒテンシュタイン	SG	シンガポール
AL	アルバニア	FI	フィンランド	LK	スリ・ランカ	SI	スロヴェニア
AM	アルメニア	FR	フランス	LR	リベリア	SK	スロヴァキア
AT	オーストリア	GA	ガボン	LS	レソト	SL	シエラ・レオネ
AU	オーストラリア	GB	英国	LT	リトアニア	SN	セネガル
AZ	アゼルバイジャン	GD	グレナダ	LU	ルクセンブルグ	SZ	スワジランド
BA	ボスニア・ヘルツェゴビナ	GE	グルジア	LV	ラトヴィア	TD	チャード
BB	バルバドス	GH	ガーナ	MC	モナコ	TG	トーゴ
BE	ベルギー	GM	ガンビア	MD	モルドヴァ	TJ	タジキスタン
BF	ブルキナ・ファソ	GN	ギニア	MG	マダガスカル	TM	トルクメニスタン
BG	ブルガリア	GW	ギニア・ビサウ	MK	マケドニア旧ユーゴスラヴィア共和国	TR	トルコ
BJ	ベナン	GR	ギリシャ	ML	マリ	TT	トリニダード・トバゴ
BR	ブラジル	HR	クロアチア	MN	モンゴル	UA	ウクライナ
BY	ベラルーシ	HU	ハンガリー	MR	モーリタニア	UG	ウガンダ
CA	カナダ	ID	インドネシア	MW	マラウイ	US	米国
CG	コンゴ	IE	アイルランド	MX	メキシコ	UZ	ウズベキスタン
CH	スイス	IL	イスラエル	NE	ニジェール	VN	ヴェトナム
CI	コートジボワール	IN	インド	NL	オランダ	YU	ユーゴスラビア
CM	カメルーン	IS	アイスランド	NO	ノルウェー	ZA	南アフリカ共和国
CN	中国	IT	イタリア	NZ	ニュージーランド	ZW	ジンバブエ
CU	キューバ	JP	日本	PL	ポーランド		
CY	キプロス	KE	ケニア	PT	ポルトガル		
CZ	チェコ	KG	キルギスタン	RO	ルーマニア		
DE	ドイツ	KP	北朝鮮	RU	ロシア		
DK	デンマーク	KR	韓国	SD	スーダン		
EE	エストニア	KZ	カザフスタン	SE	スウェーデン		
		LC	セントルシア				

明細書

ディジタル信号処理方法及び装置、制御データ生成方法及び装置、並びにプログラム記録媒体

技術分野

本発明は、楽器音情報を複数蓄積した音源から出力される演奏音信号と、この演奏音信号以外のディジタル信号とを同期再生するためのディジタル信号処理方法及び装置、上記同期再生を可能にするための制御データを生成する制御データ生成方法及び装置、並びに上記同期再生を可能にするプログラムを記録しているプログラム記録媒体に関する。

背景技術

楽器音を蓄積した音源に制御データを与え、音源から楽器の演奏音を発音させる楽器演奏制御のためのインターフェースとして、ミュージカル・インストロメント・ディジタル・インターフェース (Musical Instrument Digital Interface: MIDI) が広く用いられている。現在では、電子楽器を外部から制御するための標準インターフェースとして位置付けられている。

MIDI信号は、これに対応した電子楽器などの演奏パラメータを、ディジタル符号化したものであり、符号を修正することで復号後の演奏を修正することができる。MIDI信号の記録、編集、再生などは、

シーケンサ、またはシーケンスソフトウェアで行われ、ここでMIDI信号は、MIDIファイルとして扱われる。

また、異なるシーケンサ、あるいはシーケンスソフトウェア間でのMIDIファイルの互換性を保つ統一規格として、標準MIDIファイル (Standard MIDI File:SMF) が知られている。このSMFは、“チャンク”と呼ばれるデータ単位によって構成されている。この“チャンク”は、ヘッダ・チャンク (Header Chunk) とトラック・チャンク (Track Chunk) と呼ばれるデータが定義されている。ヘッダ・チャンクは、SMFファイルの先頭に設定され、そのファイルのデータに関する基本的な情報が記述されている。トラック・チャンクは、時間情報 (Delta-Time) とイベント (Event) で構成される。このイベントは、データファイルの各項目に変更を与えるような出来事、事象を表す。SMFフォーマット化されたMIDIファイルデータのイベントとしては大きく分けて、ミディ・イベント (MIDI Event)、システム・エクスクルーシブ・イベント (SysEx Event)、そしてメタ・イベント (Meta Event) の3種類がある。

ミディ・イベントには、演奏データそのものが示されている。システム・エクスクルーシブ・イベントには、主にMIDIのシステム・エクスクルーシブ・メッセージが示されている。システム・エクスクルーシブ・メッセージは、特定の楽器にしかない情報を交換したり、特別の非音楽情報、イベント情報などを伝達するために用いられる。メタ・イベントには、テンポや拍子などの演奏全体の情報や、シーケンサやシーケンスソフトが利用する歌詞や著作権情報などの付加的な情報が示されている。全てのメタ・イベントは0xFFで始まり、次にイベント・タイプを表すバイトが続き、更にデータ長及び

データ自体が続く。MIDI演奏プログラムは、自分が認識できないメタ・イベントを無視するように設計されている。

また、各イベントには、そのイベントを実行する時間的タイミングに関するタイミング情報が付加されている。このタイミング情報は、直前のイベントの実行からの時間差で示されている。例えば、このタイミング情報が“0”のときは、直前のイベントと同時にこのタイミング情報が付加されたイベントが実行されることになる。

一般に、MIDI規格を用いた音楽再生では、各種信号、楽器固有の音色をモデル化し、そのデータを格納した音源を各種パラメータで制御する方式をとっているため、例えば人の声や自然音など、現状でモデル化の困難な、あるいは研究が進んでいない音を表現するのは困難となっている。従って、MIDI規格による音楽の再生は、楽器の演奏程度のものとなり、例えば歌声のようなものは含まれないことがほとんどである。

そこで、MIDI規格での音楽表現の不足部分を補う新しい技術として、人の声のような、演奏音ではないオーディオ信号と、MIDI信号による演奏音との同期再生技術が求められるようになった。

ところで、従来、上記MIDI信号による演奏音と、上記演奏音ではない例えばボーカルのようなオーディオ信号が同期するシーケンスソフトウェアのようなシステムは存在しているが、そのシステムを用いる場合でしか再現できないような、複雑で、拡張性の低いものであった。

また、同期の対象としては、上記演奏音でないオーディオ信号だけでなく、MIDI信号と画像信号、MIDI信号とテキスト信号などの同期再生も考えられ、統合メディアへの拡張が期待される。また、近

年では、ネットワークを介したデータのやり取りが頻繁に行われるようになり、上述の統合メディアに対しても例外ではない。従って、ネットワークを介したシステムにおいても、この同期再生が可能となるような、扱いやすく、拡張性の高い技術が必要となる。また、MIDIデータのように、修正が容易に行えるものが望ましい。

発明の開示

本発明は、上記楽器演奏のためのインターフェースデータに記述された制御データを使って、上記音源再生には影響を及ぼすことなく、演奏音の信号とそれ以外のデジタル信号との同期再生を実現するデジタル信号処理方法及び装置の提供を目的とする。

また、本発明は、上記演奏音信号とそれ以外のデジタル信号を同期させるための制御データを含んだインターフェースデータを簡単に生成することのできる制御データ生成方法及び装置の提供を目的とする。

さらに、本発明は、上記演奏音信号とそれ以外のデジタル信号との同期再生を、適切な装置されればどこでも誰にでも可能とするソフトウェアとしてのプログラム記録媒体の提供を目的とする。

このため、本発明に係るデジタル信号処理方法は、上記課題を解決するために、楽器音情報を複数蓄積した音源に楽器の演奏音を発音させるための演奏データを少なくとも含んでいる、楽器演奏のためのインターフェースデータに基づいた演奏音信号と、上記演奏音信号以外のデジタル信号とを、上記インターフェースデータ中に予め符号化されて記述されている制御データに基づいて再生する。

具体的には、演奏音信号ではないオーディオ信号や画像信号、さらには文字信号などのデジタル信号の再生タイミングや、再生パラメータなどを、上記インターフェースデータに予め記述されている制御データで制御するものである。

この制御データは、SMFフォーマット化されたMIDIデータのイベントのうちのメタ・イベントのシーケンス・スペシフィック・イベント (Sequencer Specific Event) 上に記述されている。このシーケンス・スペシフィック・イベントには、シーケンサの製造者が独自のデータを書き込める。制御データはシーケンス・スペシフィック・イベント上で、SMFの他のイベントと同様に、容易に扱うことができる。

このシーケンス・スペシフィック・イベント上に記述される制御データは、制御内容を示す制御イベントの種類を示すIDコードと、その制御量／制御方法を組み合わせた構成とされる。

また、制御の対象となるオーディオ信号や画像信号などのデータに、データ毎に番号などのIDを持たせることで、複数のデータが存在する場合も、任意のデータを制御することが可能となる。このIDは、データにヘッダ部をもたせる等の方法で、データに付加することができる。

このため、制御データの中に、制御の対象とするデータのIDコードを含めてもよい。また、制御の対象とする信号が、オーディオ信号なのか、画像信号なのか、あるいはテキストデータなのかを示す制御信号の種類を示すIDコードを含めても良い。

これらのIDなどの制御データとして記録されるデータは、すべて簡単なビット列で表され、MIDI規格のデータと同様に、修正等を容

易に行うことができる。

また、本発明に係るデジタル信号処理装置は、上記課題を解決するために、楽器音情報を複数蓄積した音源に楽器の演奏音を発音させるための演奏データを少なくとも含んでいる、楽器演奏のためのインターフェースデータ中に予め符号化されて記述されている制御データを復号する第1の復号手段と、上記復号手段で復号した制御データに基づいて上記演奏データの再生タイミング情報に対応させて上記演奏音の信号以外のデジタル信号を復号する第2の復号手段とを備える。

また、本発明に係る制御データ生成方法は、上記課題を解決するために、楽器音情報を複数蓄積した音源から出力される演奏音信号に対して、上記演奏音信号以外のデジタル信号を同期させるための制御データを含んだインターフェースデータを生成する。

また、本発明に係る制御データ生成装置は、上記課題を解決するために、楽器音情報を複数蓄積した音源から出力される演奏音信号に対して、上記演奏音信号以外のデジタル信号を同期させるための制御データを含んだインターフェースデータを生成する手段を備えてなる。

また、本発明に係るプログラム記録媒体は、上記課題を解決するために、楽器音情報を複数蓄積した音源に楽器の演奏音を発音させるための演奏データを少なくとも含んでいる、楽器演奏のためのインターフェースデータ中に予め符号化されて記述されている制御データを復号するステップと、上記ステップで復号した制御データに基づいて上記演奏データの再生タイミング情報に対応させて上記演奏音の信号以外のデジタル信号を復号するステップとを備えてな

るソフトウェアプログラムを記録している。

図面の簡単な説明

図 1 は本発明に係るデジタル信号処理方法及び装置の実施の形態となるデジタル信号処理装置の構成を示すブロック図である。

図 2 は上記デジタル信号処理装置に供給される SMF データのメタ・イベントのフォーマット図である。

図 3 は上記図 2 に示したメタ・イベントのシーケンス・スペシフィック・イベントを示すフォーマット図である。

図 4 は上記図 3 に示したシーケンス・スペシフィック・イベントに記述される制御データのフォーマット図である。

図 5 は上記図 4 のフォーマット図で示した制御データを含む SMF データが供給されたときの上記デジタル信号処理装置の動作を説明するためのフローチャートである。

図 6 A は無音部分を含むオーディオデータを示す図である。

図 6 B は上記図 6 A の無音部分を切り出した複数の素片データを示す図である。

図 7 は複数チャンネル存在するオーディオデータを示す図である。

図 8 は制御対象データの ID を付加した制御データのフォーマット図である。

図 9 は上記図 8 にフォーマットを示した制御データを含む SMF データが供給されたときの上記デジタル信号処理装置の動作を説明するためのフローチャートである。

図 10 は上記図 8 にフォーマットを示した制御データの具体例を

示す図である。

図 1 1 は上記図 8 にフォーマットを示した制御データの他の具体例を示す図である。

図 1 2 は制御信号の種類を示す ID を付加した制御データのフォーマット図である。

図 1 3 は上記図 1 2 にフォーマットを示した制御データを含む SMF データが供給されたときの上記デジタル信号処理装置の動作を説明するためのフローチャートである。

図 1 4 は制御対象となる制御信号の種類を、ID からではなく、データ形式や内容から判断するときの上記デジタル信号処理装置の動作を説明するためのフローチャートである。

図 1 5 は本発明のプログラム記録媒体となる ROM からソフトウェアプログラムを取り出して実行する CPU を中心としたデジタル信号処理システムの構成を示すブロック図である。

図 1 6 は上記図 1 に示したデジタル信号処理装置、上記図 1 6 に示したデジタル信号処理システムで取り扱う、SMF データを生成するエンコーダの具体例を示す図である。

発明を実施するための最良の形態

以下、本発明の実施の形態について図面を参照しながら説明する。先ず、デジタル信号処理方法及び装置の実施の形態は、楽器演奏制御のためのインターフェースとして広く用いられている、ミュージカル・インストロメント・デジタル・インターフェース (Musical Instrument Digital Interface : MIDI) フォーマットの MIDI 信

号による演奏音信号と、この演奏音信号以外のデジタル信号とを同期再生するデジタル信号処理装置である。

すなわち、楽器音情報を複数蓄積したMIDI音源に楽器の演奏音を発音させるための演奏データを少なくとも含んでいる、楽器演奏のためのインターフェースデータに基づいた演奏音信号と、この演奏音信号以外のデジタル信号とを、上記インターフェースデータ中に予め符号化されて記述されている制御データに基づいて再生する本発明に係るデジタル信号処理方法を実現する装置である。

特に、以下で、このデジタル信号処理装置は、上記演奏音信号以外のデジタル信号として、人の声によるボーカル用のオーディオ信号を具体例とする。一般的にはシーケンサと呼ばれている装置と同様のものである。なお、演奏音信号と同期再生させるのは、オーディオ信号のみではなく、画像信号や、文字信号等であってもよい。

図1には上記実施の形態となるデジタル信号処理装置10の構成を示す。このデジタル信号処理装置10には、例えばインターネットを介して、エンコーダ側から伝送されてきた標準MIDIファイル(SMF)データと、上記ボーカル用のオーディオ信号が供給される。

ここで、SMFデータは、異なるシーケンサ、またはシーケンスソフトウェア間でのMIDIファイルの互換性を保つ統一規格に基づいたデータである。このSMFデータは、MIDI用の音源を直接操作するのではなく、例えばシーケンスソフトウェアを操作して、シーケンスソフトウェアにMIDI信号を出させる。そして、このMIDI信号によりMIDI音源が演奏音信号を出力する。

上記デジタル信号処理装置10において、上記SMFデータはデー

タ復号部 1 1 に取り込まれる。データ復号部 1 1 は上記 SMF データの中に後述するエンコーダ側で配置されたオーディオ制御データを抽出し、オーディオ復号部 1 2 に送る。また、データ復号部 1 1 は、上記 SMF データからパラメータ形式の MIDI データを抽出し時系列上の MIDI 信号に換えて MIDI 音源 1 3 に送る。

オーディオ復号部 1 2 は、データ復号部 1 1 からのオーディオ制御データを受け、それに従い、上記オーディオ信号を制御し、例えば、ボーカル用のオーディオ信号を再生する。

SMF データ内のオーディオ制御データは、オーディオ信号のパラメータを制御するためのデータであり、例えば、制御内容を示す制御イベントの種類を示す ID とその制御量／制御方法から構成される。このオーディオ制御データは、エンコーダ側で上記 SMF データに記述されている。

このオーディオ制御データは、上記 SMF フォーマット化された MIDI データのイベントのうちのメタ・イベントのシーケンス・スペシフィック・イベント (Sequencer Specific Event) 上に記述されている。このシーケンス・スペシフィック・イベントには、シーケンサの製造者が独自のデータを書き込める。制御データはシーケンス・スペシフィック・イベント上で、SMF の他のイベントと同様に、容易に扱うことができる。シーケンス・スペシフィック・イベントには、メタイベントを示すコード 0xFF に続くイベントタイプとして 0x7F のコードが割り当てられている。

上記オーディオ制御データの制御内容を示す制御イベントは、上記 ID により簡単なビット列で表すことができる。また、その制御量／制御方法も同様に、簡単なビット列で表すことができる。例えば、

表1に示すように定義すれば良い。

【表 1】

制御 イベントID	制御量／制御方 法を示すビット列	制御内容	デフォルト 値
0x1	0x0/0x1	Start(0x0)/Stop(0x1)	0x0
0x2	0x000～0xffff	音量コントロール	0x000
0x3	0x800～0x000～ 0x7ff(2の補数)	ピッチコントロール	0x000
0x4	0x800～0x000～ 0x7ff(2の補数)	スピードコントロール	0x000
0x5	0x0～0xf +0x00～0xff	エコーなどの効果の種 類とそのパラメータ	0x000
0x6	0x0/0x1	順方向再生／逆方向再 生	0x0
0x7	0x000～0xffff	設定音量までの到達時 間（フェードイン）	0x000
0x8	0x000～0xffff	再生後、強制停止する までの時間（但し0xffff は強制停止なしを示 す）	0xffff
0xf	0x000～0xffff	この制御の実行タイミ ング（X μ 秒後）	0x000

上記メタ・イベントのシーケンス・スペシフィック・イベントに
記述された表 1 に示すオーディオ制御データにおいて、制御イベン

トのIDが0x1で、制御量／制御方法を示すビット列が0x0のときは「オーディオ信号の再生開始」を規定している。また、制御イベントのIDがそのままで、制御量／制御方法を示すビット列が0x1のときは「オーディオ信号の再生停止」を規定している。

また、制御イベントIDが0x2である場合の制御内容は音量コントロールであり、制御量／制御方法を示すビット列が0x000～0xffffであるので(000000000000)から(111111111111)段階の音量制御を規定している。

また、制御イベントIDが0x3である場合の制御内容はピッチコントロールであり、0x800～0x000～0x7ff、つまり0x000(=000000000000)から正方向に0x800(=100000000000)段階、負方向に0x7ff(2の補数、=011111111111)段階のピッチ制御を規定している。また、制御イベントIDが0x4である場合には、同様の制御量のスピードコントロールを規定している。

また、制御イベントIDが0x5である場合、制御内容はエコーなどの効果であり、具体的に制御量／制御方法の0x0～0xfと+0x000～0xffffで効果の種類とそのパラメータを規定している。また、制御イベントIDが0x6である場合には、順方向再生／逆方向再生を、制御量／制御方法が0x0/0x1であることで規定している。

また、制御イベントIDが0x7である場合には、設定音量までのフェードインによる到達時間を示し、その制御量は0x000～0xffffであることを規定している。また、制御イベントIDが0x8である場合には、再生後、強制停止するまでの時間を示し、0x000～0xffffの制御量を規定している。

そして、制御イベントIDが0xfである場合には、その制御を実際に

行うタイミングを0x000～0xffffの制御量で規定している。ここでの単位は μ 秒であり、具体的には μ 秒後に実行されることを規定している。

SMFの各イベントでは、その発音や制御をリアルタイムの時間的タイミングで記述しているが、本発明で用いる上記制御データ中の制御イベントでは、実際の制御を別の時間的タイミングで実行するように定義できる。すなわち、上記表1で示した制御イベントID“0xf”で設定される実行タイミングは、トラック・チャンクのメタ・イベントに設定された時間情報と、実際に制御を実行する時間的タイミングとの相対的な時間をビット列で示している。また、全体の再生における絶対的な時間（MIDIシーケンスの演奏開始時間からの時間）をビット列で示すことなども考えられる。

ところで、表1において制御データを構成する制御イベントや、制御量／制御方法を示すのに必要なビット列の長さは、制御の内容（種類）によって異なっている。例えば、オーディオ信号の再生および停止は1ビットの情報量でも表現可能であるが、音量などは用途に応じて1～数ビットの情報量が必要となる。逆に言えば、オーディオ信号の再生／停止を表すために、数バイトも使うのは冗長である。従って、上記表1に示した制御データの内の制御量／制御方法を示すビット列の長さは、全ての制御において固定で等しいビット長を割り当てるよりも、制御の種類（制御イベントID）ごとに定義するほうが冗長性が低くなる。また、各種制御の制御量／制御方法の生成確率に応じて情報を可変長符号化しても良い。

このような制御データからなるメタ・イベントの具体例について以下に説明する。まず、SMFのトラック・チャンクは、上述したよう

に、時間情報とイベントで構成され、さらにイベントには、ミディ・イベント (MIDI Event)、システム・エクスクルーシブ・イベント (SysEx Event)、そしてメタ・イベント (Meta Event) の3種類があった。各イベントには可変長の時間データが頭の部分に付加されている。この時間データを取り除いたときの、メタ・イベントの構成を図2に示す。

左側の1バイト [FF] でメタ・イベントであることを表し、次の1バイトでメタ・イベントにおけるMIDI音源制御のためのテンポや拍子やキーなどの演奏全体の情報等を [イベントの種類] として示す。また、次の [データ長] では、メタ・イベントデータの長さをバイト単位で示す。残りが実際のメタ・イベントデータの内容である。

このメタ・イベントにおいて実際のオーディオ信号を制御するためのオーディオ制御データは、図3に示すように上記 [イベントの種類] が [7F] となる、シーケンス・スペシフィック・イベント (Sequencer Specific Event) に記述される。このシーケンス・スペシフィック・イベントには、シーケンサの製造者が独自のデータを書き込める。

すなわち、先頭が [FF] であり次の [イベントの種類] が [7F] であるときにはメタ・イベント内のシーケンス・スペシフィック・イベントであることを示す。

例えば、図3に示したシーケンス・スペシフィック・イベントの、[メーカーID] の直後に、図4に示すように [オーディオ制御データ] を並べて記録することにより、複数のオーディオ制御を一つのシーケンス・スペシフィック・イベントで指示、また同時に実行す

ることができる。なお、[オーディオ制御データ]は上記表1に示したように、[制御イベントID]と[制御量/制御方法]からなるが、図4には[制御イベントのID]と簡略化のため[制御量]とのみ記している。

また、複数のシーケンス・スペシフィック・イベントで、複数の制御を指示して、それらを同時に行うことも可能である。この制御データとして、上記表1に示したオーディオ制御データとしての[制御イベントID]及び[制御量/制御方法を示すビット列]を記述することにより、オーディオ信号を制御できる。

なお、図3及び図4に示すように、メタ・イベント内には製造者を識別するためのメーカIDが記述されている。これは、シーケンス・スペシフィック・イベントに独自のデータを記述した製造者を識別するものであり、製造者によって記述される。このメーカIDによって識別される製造者、又はその許可を受けた製造者によって製作された装置やプログラムソフトウェアにより、上記シーケンス・スペシフィック・イベントを復号することにより上記演奏音信号とそれ以外のボーカルオーディオ信号を同期再生することが可能となる。上記メーカIDを識別できない装置やソフトウェアにおいては、上記シーケンス・スペシフィック・イベントに記述された内容は無視される。したがって、シーケンサやシーケンスソフトが本発明に対応していない場合であっても、MIDI信号再生は通常どおり行われる。

この図4に示す構成のSMFデータと、ボーカルのオーディオ信号がエンコーダ側から送られてきたときの上記図1に示すデジタル信号処理装置10の動作について図5のフローチャートを用いて説明する。ここでは、上記デジタル信号処理装置10に後述するよう

に制御対象データが1個のみ既に読み込まれていることを前提として説明を進める。

先ず、オーディオ信号処理装置10は、データ復号部11に上記SMFデータを取り込むと、ステップS1でイベントを読み込む。そして、ステップS2で図4に示すようにSMF化されたMIDIデータのイベント種類を示す1バイトが[FF]であるか否かを判断し、[FF]であればメタ・イベントであるので、ステップS3に進む。

次に、ステップS3で[イベントの種類]を見て[7F]であるか否かを判断し、[7F]であればシーケンス・スペシフィック・イベントであるので、ステップS4に進む。そして、ステップS4でそのシーケンス・スペシフィック・イベントにオーディオ制御データが記述されているか否かを判断する。具体的にはシーケンス・スペシフィック・イベントデータにオーディオ制御データとして上記表1に示したオーディオ信号を制御するための制御イベントIDが存在するか否かを判断する。ここで、オーディオ制御データが書き込まれていると判断するとステップS5に進む。

ステップS5では、制御対象データが存在しているか否かを判断する。すなわち、オーディオ制御データとして示された制御を行う対象となる、上記ボーカル用のオーディオ信号がデジタル信号処理装置10内部に読み込みが完了しているかどうかを判断する。具体的にはネットワークなどを介して上記デジタル信号処理装置10内部に既にダウンロードされているか否かを判断する。ここで、制御対象データが存在していると判断すると、ステップS6に進んで、後述するように上記表1に示した制御の種類(内容)と制御量／制御方法を読み込む。

なお、このフローチャートにおいては、制御対象データが1個のみの場合を示しており、制御対象データが存在しなければ後述するステップS 13に進むようになっている。複数の制御対象データを扱うような処理とするには、いずれか一つでも制御対象データが存在するのであれば、その制御対象データに対して制御を行うような処理の流れとすればよい。

ステップS 6でデジタル信号処理装置10は、上記オーディオ制御データ中の〔制御イベントID〕と〔制御量／制御方法〕を読み込む。そして、ステップS 6で読み込んだ値が正しいか否かをステップS 7で判断する。

例えば、ステップS 7で、制御イベントIDが0x1であり、制御量／制御方法を示すビット列が0x0又は0x1であれば、値は正しいと判断してステップS 8に進む。一方、制御イベントIDが0x1であるにもかかわらず、制御量／制御方法が0x000～0xfffであるとき、制御の種類と制御量／制御方法が間違っていると判断してステップS 9に進む。

ステップS 8では、制御イベントIDと、制御量／制御方法を示すビット列が正しいので、例えば制御イベントIDの0x1と、制御量／制御方法の0x0に基づいてオーディオ復号部12での上記オーディオ信号の復号を設定し、ステップS 9に進む。

ステップS 9では、メタ・イベントのオーディオ制御データが終わりか否かを判断する。オーディオ制御データが複数指示されているときは、終わりでないと判断し、ステップS 6に戻り、指示されたすべての制御内容と制御量／制御方法を繰り返し読み込んで設定する。なお、メタ・イベントの終了判定は、シーケンサ・スベシフ

ック・イベントを表すコード [7 F] の後ろに位置する [データ長] に基づいて判断される。

次に、ステップ S 1 0 では、制御対象データ、すなわち、制御しようとしているオーディオ信号が現在再生中かどうかを調べる。これは次のステップ S 1 1 において、上記ステップ S 6 からステップ S 8 までの制御値の設定のループで未設定の制御値を既定値（デフォルト値）で初期化するかどうかを決定するために行われる。制御対象データが再生中であれば、ステップ S 1 2 に進み、後述するように制御を実行する。例えば、上記制御対象データが現在再生中の場合には、オーディオ制御データで指定された設定値のみを反映するようにすればよい。例えば、ピッチコントロールの制御がオーディオ制御データに記述されていた場合には、その制御値のみをコントロールすることにより、例えば、その他の制御値である音量やスピードなどを現在の再生状態に保持したままピッチを制御することになる。

一方、上記制御対象データが再生中でなければ、ステップ S 1 1 に進んで、未設定の制御値を既定値（デフォルト値）に設定する。

ステップ S 1 2 では、上記オーディオ信号に対する制御が、上記オーディオ制御データに付加された時間データで指示されたタイミングで実行される。すなわち、制御イベント ID が 0xf と記述されている場合には、その後の制御量／制御方法で示す 0x000～0xffff の範囲の $\times \mu$ 秒後に、上記制御の種類が実行される。

そして、ステップ S 1 3 では、イベントのシーケンスが終了したか否か、すなわち演奏が終了してから否かを判断する。

なお、ステップ S 1 4 は、ステップ S 2 でメタ・イベントでない

と判断したとき、また、ステップ S 3 でシーケンス・スペシフィック・イベントでないと判断したとき、さらに、ステップ S 4 でシーケンス・スペシフィック・イベントにオーディオ制御データが記述されていないと判断したときに実行され、その他のイベント処理が行われる。例えば、ステップ S 3 から進んできたときには、[イベントの種類] に応じた所定のメタ・イベントが実行される。例えば、MIDIデータのテンポ、拍子などを制御する。

一方、上記データ復号部 11 は、上記 SMF データ中から復号した MIDI 信号を MIDI 音源 13 に送り、MIDI 音源 13 を駆動させ演奏音信号に応じた音を発音させる。

オーディオ復号部 12 は、データ復号部 11 で復号されたオーディオ制御データを受け、それにしたがって、オーディオ出力を発音させる。このオーディオ復号部 12 での復号時間が十分に短ければ、上記オーディオ制御データは SMF データのメタ・イベントから復号したものであるので、上記演奏音とオーディオ出力は同期のとれた再生音となる。

このように、上記図 4 に示すようなフォーマットのメタ・イベントを含む SMF データが送られてきたとき、上記図 1 に示したデジタル信号処理装置 10 は、単に MIDI 信号に基づいた上記演奏音を再生するだけでなく、MIDI 規格での音楽再生では再現が困難なボーカルや自然音などを、上記演奏音に同期させて再生することができる。

なお、ここで、MIDI 信号を再生しようとするシーケンサが上記デジタル信号処理装置 10 に対応していない場合、上記 SMF データ上の制御イベントは無視されるので、MIDI データの互換性は保たれる。

次に制御対象となるデータが複数ある場合であって、制御の対象

とするオーディオ信号を示すIDを上記オーディオ制御データに付加した場合について説明する。連続する制御対象データを一塊りとしてIDを付加することにより、複数のオーディオ信号が用意されている場合でも、特定のオーディオ信号を制御することが可能となる。つまり、制御データ中の制御イベントで指示される制御は、上記制御対象データIDをもつ制御対象データに対してのみ実行される。

例えば、図6Aに示すように、斜線部で示す無音部分を含むボカルのオーディオ信号があるとする。このオーディオ信号を図6Bのように、無音部分を切り出し、複数の素片に分けることで、全体の信号量を減らすことが可能である。

このようにしてできた複数の素片データ、もしくは、図7のようにオーディオ信号が複数チャンネル存在するときなどのように、複数個の制御対象データをそれぞれ独立して制御もしくは取り扱うために、図8に示す〔制御対象データのID〕を用いる。

これに対して、オーディオ信号の、各素片データにも番号などのIDを持たせる。このIDは、それぞれの信号の一部として持たせるなどする。そして、信号を再生する等の制御を行うときには、このIDを上記〔制御対象データのID〕として指定して制御することで、複数のオーディオ信号が存在する場合も、任意の信号を制御することが可能となる。

このように、制御の対象となるオーディオ信号や画像信号などに、連続するデータ毎に番号等のIDを持たせることで、複数の信号ブロックが存在する場合も、任意の信号ブロックを制御することが可能となる。このIDは、信号ブロックにヘッダ部を持たせる等の方法で、付加することができる。

このため、制御データの中に、制御の対象とするデータのIDコードを含めてもよい。また、後述するように、制御の対象とする信号が、オーディオ信号なのか、画像信号なのか、あるいは文字信号のようなテキストデータなのかを示す制御信号の種類を示すIDコードを含めてもよい。

この図8に示した構成のSMFデータと、ボーカルのオーディオ信号がエンコーダ側から送られてきたときの上記図1に示すデジタル信号処理装置10の動作について図9のフローチャートを用いて説明する。

先ず、オーディオ信号処理装置10は、データ復号部11に上記SMFデータを取り込むと、ステップS21でイベントを読み込む。そして、ステップS22でMIDIデータのイベント種類を示す1バイトが[FF]であるか否かを判断し、[FF]であればメタ・イベントであるので、ステップS23に進む。

次に、ステップS23で[イベントの種類]を見て[7F]であるか否かを判断し、[7F]であればシーケンス・スペシフィック・イベントであるので、ステップS24に進む。そして、ステップS24でそのシーケンス・スペシフィック・イベントにオーディオ制御データが記述されているか否かを判断する。ここで、オーディオ制御データが書き込まれていると判断するとステップS25に進む。

ステップS25では、制御対象IDを持つデータが存在しているか否かを判断する。ここでの判断処理は、制御対象データが上記図6の(b)や、図7に示すように、複数存在するときに適用されるものである。制御対象データのIDを持つデータが存在していると

判断すれば、ステップS 2 6以降を実行する。

ステップS 2 6で、デジタル信号処理装置10は、上記オーディオ制御データ中の〔制御イベントID〕と〔制御量／制御方法〕を読み込む。なお、データ復号部11は、ステップS 2 6で読み込んだ値が正しいか否かをステップS 2 7で判断する。

ステップS 2 8では、ステップS 2 7での判断結果、すなわち制御イベントIDと、制御量／制御方法を示すビット列が正しいとの判断結果を受けて、オーディオ復号部12での上記オーディオ信号の復号を設定し、ステップS 2 9に進む。

ステップS 2 9では、メタ・イベントのオーディオ制御データが終わりか否かを判断する。オーディオ制御データが複数指示されているときは、終わりでないと判断し、ステップS 2 6に戻り、指示されたすべての制御内容と制御量／制御方法を繰り返し読み込んで設定する。

次に、ステップS 3 0では、制御対象データが現在再生中かどうかを調べる。これは次のステップS 3 1において、上記ステップS 2 6からステップS 2 8までの制御値の設定のループで未設定の制御値を既定値（デフォルト値）で初期化するかどうかを決定するために行われる。制御対象データが再生中であれば、ステップS 3 2に進み、後述するように制御を実行する。

一方、上記制御対象データが再生中でなければ、ステップS 3 1に進んで、未設定の制御値を既定値（デフォルト値）に設定する。

ステップS 3 2では、上記オーディオ信号に対する制御が、上記オーディオ制御データに付加された時間データで指示されたタイミングで実行される。

そして、ステップ S 3 3 では、イベントのシーケンスが終了したか否か、すなわち演奏が終了してから否かを判断する。

なお、ステップ S 3 4 は、ステップ S 2 2 でメタ・イベントでないと判断したとき、また、ステップ S 2 3 でシーケンス・スペシフィック・イベントでないと判断したとき、さらに、ステップ S 2 4 でシーケンス・スペシフィック・イベントにオーディオ制御データが記述されていないと判断したときに実行され、その他のイベント処理が行われる。

例えば、図 8 の具体例として図 1 0 のような SMF データがデジタル信号処理装置 1 0 に送られてきたとき、データ復号部 1 1 は〔制御の対象の ID 番号〕が〔0 1〕であるので、図 6 の（b）又は図 7 に示すようなオーディオデータの素片の ID〔0 1〕を制御の対象とすることを復号し、また〔制御イベントの ID〕が〔1〕で〔制御方法〕が〔0〕であるので、上記表 1 に基づいてオーディオの再生開始を復号し、また〔制御イベント〕が〔2〕で〔制御量〕が〔0 1 E〕であるので音量が 3 0 であることを復号する。このため、オーディオ復号部 1 2 には制御対象の ID〔0 1〕のオーディオデータを、音量 3 0 で再生をしろというオーディオ制御データが供給される。ここで、これらの内容以外の制御は既定値（デフォルト値）を参照することによって行われる。

また、MIDI 信号再生のピッチやスピードを変更した場合、図 1 1 に示す SMF データを用いることで、その変更に従って、オーディオ信号再生を行うことができる。例えば、ピッチが元のピッチに対して 2 度高くなり、MIDI 信号再生のスピードがもとのスピードに対して 1 0 % 遅くする場合の制御イベントである。

この場合、表 1 に従うとすると、制御対象のデータは ID 番号 1 のオーディオ信号であり、オーディオ信号のピッチを 2 度高く、また再生スピードを 10 % 遅くさせ、上記以外の制御を既定値（制御なし）に設定させる。再生開始のタイミングだけでなく、ピッチやスピードなどに対しても、上記演奏音信号と同期させることができる。

なお、制御の対象がオーディオ信号以外の画像信号やテキスト信号などの場合も同様に実行できる。また、オーディオ信号と画像信号を制御するというように、複数の MIDI 信号以外のデジタル信号を制御することも可能である。

この場合、図 12 に示すように、オーディオ信号や画像信号などの、制御されるデジタル信号が何かを示す〔制御信号の種類を示す ID〕が制御イベントに記述され、この ID を分類別けすることにより、制御イベントの拡張性を高める方法が採られる。つまり、「オーディオ信号は 0x01」などの様に示すことで、制御の対象となるデジタル信号が複数種類あるとき、また今後新しい種のデジタル信号が加えられたとき、その対応が容易となる。

この図 12 に示す SMF データがエンコーダ側から送られてきたときの上記デジタル信号処理装置 10 の動作について図 13 を用いながら説明する。この場合、上記デジタル信号処理装置 10 は、上記シーケンサとしての用途だけで動作するのではないことはいうまでもない。また、上記オーディオ復号部 12 は、単に復号部 12 という名称に変更する。

まず、オーディオ信号処理装置 10 は、データ復号部 11 に上記 SMF データを取り込むと、ステップ S41 でイベントを読み込む。そ

して、ステップS 4 2でMIDIデータのイベント種類を示す1バイトが〔F F〕であるか否かを判断し、〔F F〕であればメタ・イベントであるので、ステップS 4 3に進む。

次に、ステップS 4 3で〔イベントの種類〕を見て〔7 F〕であるか否かを判断し、〔7 F〕であればシーケンス・スペシフィック・イベントであるので、ステップS 4 4に進む。そして、ステップS 4 4でそのシーケンス・スペシフィック・イベントにオーディオ制御データが記述されているか否かを判断する。ここで、オーディオ制御データが書き込まれていると判断するとステップS 4 5に進む。

ステップS 4 5では、制御対象データが存在しているか否かを判断する。制御対象データが存在していると判断すれば、ステップS 4 6以降を実行する。

ステップS 4 6で、デジタル信号処理装置10は、上記オーディオ制御データ中の〔制御される信号の種類を示すID〕を読み込む。なお、デジタル信号処理装置10はステップS 4 6で読み込んだIDが処理可能なID種類であるか否かをステップS 4 7で判断する。すなわち、読み込んだIDがオーディオ信号、画像信号、あるいはテキストデータ等制御可能なものであるのか否かを判断する。ここで、処理可能なIDであると判断すれば、ステップS 4 8に進むが、処理可能なIDでは無いと判断すると、ステップS 5 5にとぶ。

次に、デジタル信号処理装置10は、ステップS 4 8で上記オーディオ制御データ中の〔制御イベントID〕と〔制御量／制御方法〕を読み込む。そして、ステップS 4 8で読み込んだ値が正しいか否かをステップS 4 9で判断する。

ステップS 4 9で制御イベントIDと、制御量／制御方法を示すビット列が正しいと判断すると、ステップS 5 0に進んで、オーディオ復号部1 2での上記オーディオ信号の復号を設定し、さらにステップS 5 1に進む。ステップS 4 9で制御イベントIDと、制御量／制御方法を示すビット列が正しくないと判断するとステップS 5 1に進む。

ステップS 5 1では、メタ・イベントのオーディオ制御データが終わりか否かを判断する。オーディオ制御データが複数指示されているときは、終わりでないと判断し、ステップS 4 8に戻り、指示されたすべての制御内容と制御量／制御方法を繰り返し読み込んで設定する。

次に、ステップS 5 2では、制御対象データが現在再生中かどうかを調べる。これは次のステップS 5 3において、上記ステップS 4 8からステップS 5 0までの制御値の設定のループで未設定の制御値を既定値（デフォルト値）で初期化するかどうかを決定するために行われる。制御対象データが再生中であれば、ステップS 5 4に進み、後述するように制御を実行する。

一方、上記制御対象データが再生中でなければ、ステップS 5 3に進んで、未設定の制御値を既定値（デフォルト値）に設定する。

ステップS 5 4では、上記オーディオ信号に対する制御が、上記オーディオ制御データに付加された時間データで指示されたタイミングで実行される。

そして、ステップS 5 5では、イベントのシーケンスが終了したか否か、すなわち演奏が終了してから否かを判断する。

なお、ステップS 5 6は、ステップS 4 2でメタ・イベントでな

いと判断したとき、また、ステップS 4 3でシーケンス・スペシフィック・イベントでないと判断したとき、さらに、ステップS 4 4でシーケンス・スペシフィック・イベントにオーディオ制御データが記述されていないと判断したときに実行され、その他のイベント処理が行われる。

一方、データ復号部 1 1は、上記SMFデータ中のMIDI信号をMIDI音源 1 2に送り、MIDI音源 1 3を駆動させ、MIDI信号によって制御された演奏音を発音させる。

復号部 1 2は、データ復号部 1 1で復号されたオーディオ制御データを受け取り、それにしたがって、オーディオ出力、画像データ出力又はテキストデータ出力を復号する。この復号部 1 2での復号時間が十分に短ければ、上記制御データはSMFデータのメタ・イベントから復号したものであるので、上記演奏音であるMIDI出力とこの復号部 1 2からの出力は同期がとられることになる。

このように、上記図 1 2に示すようなフォーマットのメタ・イベントを含むSMFデータが送られてきたとき、上記デジタル信号処理装置 1 0は、単にMIDI信号を再生するだけでなく、MIDI規格での音楽再生では再現が困難なボーカルや自然音などを、デジタルオーディオ信号により同期再生することができる。また、制御対象のデジタル信号を示すIDにより、制御の種類を分類別けでき、拡張性をもたせることが容易となる。このため、MIDI信号の再生にあわせて、動画像や静止画像、テキスト情報をディスプレイに表示させるなども可能となる。

また、制御対象のデータが決まれば、データ形式や内容から、そのデータが、オーディオや画像など、どの種のデジタル信号のも

のか判るので、それにより、制御種類が、どの種のデジタル信号に属するかを判断でき、前述と同様に、制御の種類を分類別けすることも可能である。

このときの上記デジタル信号処理装置 10 の動作を図 14 に示すフローチャートを用いて説明する。このフローチャートは、上記図 13 に示したステップ S 46, ステップ S 47 をステップ S 60, ステップ S 61 及びステップ S 62 に置き換えてなる。

まず、オーディオ信号処理装置 10 は、データ復号部 11 に上記 SMF データを取り込むと、ステップ S 41 でイベントを読み込む。そして、ステップ S 42 で MIDI データのイベント種類を示す 1 バイトが [FF] であるか否かを判断し、[FF] であればメタ・イベントであるので、ステップ S 43 に進む。

次に、ステップ S 43 で [イベントの種類] を見て [7F] であるか否かを判断し、[7F] であればシーケンス・スペシフィック・イベントであるので、ステップ S 44 に進む。そして、ステップ S 44 でそのシーケンス・スペシフィック・イベントにオーディオ制御データが記述されているか否かを判断する。ここで、オーディオ制御データが書き込まれていると判断するとステップ S 45 に進む。

ステップ S 45 では、制御対象データが存在しているか否かを判断する。制御対象データが存在していると判断すれば、ステップ S 60 以降を実行する。

ステップ S 60 で、デジタル信号処理装置 10 は、ステップ S 45 で存在していると判断した制御対象のデータから、制御される信号が何の種類かを判別する。

そして、ステップS 6 1で判別できたか否かを判断し、判別できたのであればステップS 6 2に進んで判別できた信号が処理可能な種類か否かを判断する。ステップS 6 1、及びステップS 6 2で、制御される信号が何の種類か判別できない、及び処理可能な種類でないと判断したら、ステップS 5 5に進む。

ステップS 6 2で処理可能な種類であると判断すると、ステップS 4 8以降に進む。

ステップS 4 8で上記オーディオ制御データ中の〔制御イベントID〕と〔制御量／制御方法〕を読み込む。そして、ステップS 4 8で読み込んだ値が正しいか否かをステップS 4 9で判断する。

ステップS 4 9で制御イベントIDと、制御量／制御方法を示すビット列が正しいと判断すると、ステップS 5 0に進んで、オーディオ復号部1 2での上記オーディオ信号の復号を設定し、さらにステップS 5 1に進む。ステップS 4 9で制御イベントIDと、制御量／制御方法を示すビット列が正しくないと判断するとステップS 5 1に進む。

ステップS 5 1では、メタ・イベントのオーディオ制御データが終わりか否かを判断する。オーディオ制御データが複数指示されているときは、終わりでないと判断し、ステップS 4 8に戻り、指示されたすべての制御内容と制御量／制御方法を繰り返し読み込んで設定する。

次に、ステップS 5 2では、制御対象データが現在再生中かどうかを調べる。これは次のステップS 5 3において、上記ステップS 4 8からステップS 5 0までの制御値の設定のループで未設定の制御値を既定値（デフォルト値）で初期化するかどうかを決定するた

めに行われる。制御対象データが再生中であれば、ステップ S 5 4 に進み、後述するように制御を実行する。

一方、上記制御対象データが再生中でなければ、ステップ S 5 3 に進んで、未設定の制御値を既定値（デフォルト値）に設定する。

ステップ S 5 4 では、上記オーディオ信号に対する制御が、上記オーディオ制御データに付加された時間データで指示されたタイミングで実行される。

そして、ステップ S 5 5 では、イベントのシーケンスが終了したか否か、すなわち演奏が終了してから否かを判断する。

なお、ステップ S 5 6 は、ステップ S 4 2 でメタ・イベントでないと判断したとき、また、ステップ S 4 3 でシーケンス・スペシフィック・イベントでないと判断したとき、さらに、ステップ S 4 4 でシーケンス・スペシフィック・イベントにオーディオ制御データが記述されていないと判断したときに実行され、その他のイベント処理が行われる。

すなわち、この図 1 4 に示した処理において、デジタル信号処理装置 1 0 は、制御されるデジタル信号がどの種のものかを判断するために、前述の ID を用いずに、制御対象データの形式や情報から、それが何のデータか判別する。

以上のように、SMFデータのシーケンス・スペシフィック・イベントを用いて、オーディオ信号や画像信号などのMIDI信号以外のデジタル信号を制御することが可能となる。また、このようにすれば、MIDI規格での音楽再生への影響もなく、シーケンサ、またはシーケンスソフトが本発明に対応していない場合であっても、MIDI信号再生は通常どおり行われる。

上記実施の形態となる、デジタル信号処理装置 10 は、ハードウェアとして、上記デジタル信号処理方法を実現した。なお、上記デジタル信号処理方法を、ソフトウェアプログラムとして適用してもよい。

このデジタル信号処理方法を適用したソフトウェアプログラムは、楽器音情報を複数蓄積した音源に楽器の演奏音を発音させるための演奏データを少なくとも含んでいる、楽器演奏のためのインターフェースデータ中に予め符号化されて記述されている制御データを復号するステップと、上記ステップで復号した制御データに基づいて上記演奏データの再生タイミング情報に対応させて上記演奏音の信号以外のデジタル信号を復号するステップとを備えてなるプログラムであり、半導体メモリや、磁気ディスク、あるいは光ディスクなどの記録媒体に格納されている。

図 15 には、上記ソフトウェアプログラムを記録した ROM 22 から命令を逐次取り出して実行する CPU 20 を中心としたデジタル信号処理システムの構成を示す。CPU 20 にはバス 21 を介して ROM 22 と、ワークエリアとなる RAM 23 と、I/O インターフェース 24 が接続されている。I/O インターフェース 24 にはオーディオ信号入力端子 25 と、スピーカ 26 が接続されている。

CPU 20 は、ROM 22 から上記デジタル信号処理方法を適用したソフトウェアプログラムを逐次取り出して実行する。CPU 20 は、上記ソフトウェアプログラムを実行することにより、実質的に上記図 1 に示したデジタル信号処理装置 10 と同様の処理を行う。そして、入力端子 25 を介して供給される、例えばボーカル

のようなオーディオ信号と、MIDI音源からの演奏音信号とを同期再生し、スピーカから発音する。

このように図 1 5 に示したデジタル信号処理システムでは、ROM 2 2 に記録されている上記ソフトウェアプログラムを逐次取り出し実行することにより、今まで困難とされていた、上記演奏音信号とそれ以外のデジタル信号との同期再生を実現することができる。

つまり、ROM 2 2 に記録されている上記プログラムは上記演奏音信号とそれ以外のデジタル信号との同期再生を、適切な装置されればどこでも誰にでも可能とする。

もちろん、演奏音信号と同期再生できるのは、オーディオ信号のみではなく、画像信号や、文字信号等であってもよい。

次に、上記図 1 に示したデジタル信号処理装置 1 0、上記図 1 6 に示したデジタル信号処理システムで取り扱う、SMFデータを生成するエンコーダの具体例について図 1 6 を用いて説明する。

このエンコーダの具体例は、楽器音情報を複数蓄積した音源から出力される演奏音信号に対して、上記演奏音信号以外のデジタル信号を同期させるための制御データを含んだインターフェースデータとしての上記SMFデータを生成するための制御データ生成方法を実行するエンコーダ 3 0 である。

このエンコーダ 3 0 には、入力データとしてSMFデータ 3 1 が供給される。SMFデータ 3 1 のトラック・チャックにはミディ・イベント、システム・エクスクルーシブ・イベント、そしてメタ・イベントが記述されている。

また、エンコーダ 3 0 にはユーザによって例えばボーカルのように

なオーディオ信号と上記表 1 に示したような内容のパラメータ 3 2
1、3 2 2・・・3 2 nが入力される。これらのパラメータはMIDI音源
による演奏音信号と上記オーディオ信号とを同期再生するためのパ
ラメータである。

すると、エンコーダ 3 0 は、既に記述されているイベント間の所
定の位置に、シーケンス・スペシフィック・イベントデータに上記
制御イベントのID、制御量／制御方法からなるオーディオ制御デー
タが記述されたSMF' データ 3 3 を生成する。

このように、上記エンコーダ 3 0 によれば、上記演奏音信号とそ
れ以外のデジタル信号を同期させるためのオーディオ制御データ
を含んだSMF' データを簡単に生成することができる。

もちろん、演奏音信号と同期再生させるのは、オーディオ信号の
みではなく、画像信号や、文字信号等であってもよいので、上記制
御データとしては、画像信号や、文字信号の表示タイミングを示す
パラメータを制御するためのデータであってもよい。

なお、上記実施の形態となるデジタル信号処理装置には、例え
ばインターネット等のネットワーク媒体を介して上記SMFデータと上
記インターフェースデータ以外のデジタル信号とが供給されたが、
予めこれらの各データを例えばハードディスクドライブや光学ディ
スクドライブ等の大容量記録媒体に記録しておき、この大容量記録
媒体から上記各データを取り出して上記デジタル信号処理を行う
ようにしてもよい。

以上のように、本発明によれば、MIDI規格での音楽再生には影響
を及ぼすことなく、演奏音信号とオーディオ信号や画像信号などの
演奏音信号以外のデジタル信号との同期再生を可能とする。例え

ば、本発明に対応したシーケンサまたはシーケンスソフトは、単に演奏音信号を再生するだけでなく、MIDI規格での音楽再生では再現が困難なボーカルや自然音などを、ディジタルオーディオ信号により同期再生することができる。または、MIDI信号の再生にあわせて、動画像や静止画像、テキスト情報をディスプレイに表示させるなども可能となる。

演奏音信号も含め、これらのディジタル信号のデータは、あらかじめ記録装置、もしくは媒体上に保持しているもののほか、ネットワーク上のストリームデータを用いることも考えられる。

また、SMFのシーケンス・スペシフィック・イベントを利用することにより、MIDI再生に影響を及ぼすことなく、制御データを記録することができ、その編集も容易である。

さらに、オーディオ信号や画像信号などの再生タイミングだけでなく、各パラメータも制御できるので、演奏音信号とは独立したパラメータの変更が可能である。例えば、オーディオだけにエコーをかけることなどが可能である。

逆に、演奏音信号再生の変更に従って、オーディオや画像などのパラメータを変更することも可能である。例えば、演奏音信号による再生音楽のピッチを変更した場合、オーディオのピッチもそれに合わせて変更、再生することができる。

このように、演奏音信号とそれ以外のディジタル信号の同期再生において、容易にインタラクティブ性をもたせることができる。

また、本発明によれば、上記演奏音信号とそれ以外のディジタル信号を同期させるための制御データを含んだインターフェースデータを簡単に生成することができる。

さらに、上記演奏音信号とそれ以外のデジタル信号との同期再生を、適切な装置されあればどこでも誰にでも可能とする。

請求の範囲

1. 楽器音情報を複数蓄積した音源に楽器の演奏音を発音させるための演奏データを少なくとも含んでいる、楽器演奏のためのインターフェースデータに基づいた演奏音信号と、上記演奏音信号以外のデジタル信号とを、上記インターフェースデータ中に予め符号化されて記述されている制御データに基づいて再生することを特徴とするデジタル信号処理方法。
2. 上記演奏音信号以外のデジタル信号は、オーディオ信号であることを特徴とする請求の範囲第1項記載のデジタル信号処理方法。
3. 上記演奏音信号以外のデジタル信号は、画像信号であることを特徴とする請求の範囲第1項記載のデジタル信号処理方法。
4. 上記演奏音信号以外のデジタル信号は、文字を表す信号であることを特徴とする請求の範囲第1項記載のデジタル信号処理方法。
5. 上記制御データは、上記演奏音信号以外のデジタル信号の再生タイミングを制御するデータであることを特徴とする請求の範囲第1項記載のデジタル信号処理方法。
6. 上記制御データは、上記演奏音信号以外のデジタル信号の再生に関するパラメータを制御するデータであることを特徴とする請求の範囲第1項記載のデジタル信号処理方法。
7. 上記制御データは、上記インターフェースデータの標準フォーマットにおける所定のイベントに、符号化されたデータ列として予め記述されていることを特徴とする請求の範囲第1項記載のディ

タル信号処理方法。

８．上記制御データを制御したい時間的タイミングで記述しておくことを特徴とする請求の範囲第７項記載のデジタル信号処理方法。

９．上記制御データを制御したい時間的タイミングよりも前に記述し、上記制御データ中に実際に制御したいタイミング情報を加えておくことを特徴とする請求の範囲第７項記載のデジタル信号処理方法。

１０．上記タイミング情報を、上記制御データが記述されている時間的タイミングと実際に制御を行いたい時間的タイミングとの相対的な時間として示すことを特徴とする請求の範囲第９項記載のデジタル信号処理方法。

１１．上記タイミング情報を、全体の再生における絶対的な時間で示すことを特徴とする請求の範囲第９項記載のデジタル信号処理方法。

１２．上記演奏音信号以外のデジタル信号は、ブロック毎に識別コードを持つことを特徴とする請求の範囲第１項記載のデジタル信号処理方法。

１３．上記識別コードを指定することで特定のブロック信号を制御することを特徴とする請求の範囲第１２項記載のデジタル信号処理方法。

１４．上記制御データは、上記演奏音信号以外のデジタル信号の種類を識別コードで指定するデータであることを特徴とする請求の範囲第７項記載のデジタル信号処理方法。

１５．上記制御データは、上記演奏音信号以外のデジタル信号に対する制御内容の各制御量／制御方法をビット列で表現するデータ

であることを特徴とする請求の範囲第 7 項記載のデジタル信号処理方法。

16. 上記制御データは、上記演奏音信号以外のデジタル信号に対する複数の制御を同時に行うためのデータであることを特徴とする請求の範囲第 7 項記載のデジタル信号処理方法。

17. 楽器音情報を複数蓄積した音源に楽器の演奏音を発音させるための演奏データを少なくとも含んでいる、楽器演奏のためのインターフェースデータ中に予め符号化されて記述されている制御データを復号する第 1 の復号手段と、

上記復号手段で復号した制御データに基づいて上記演奏データの再生タイミング情報に対応させて上記演奏音の信号以外のデジタル信号を復号する第 2 の復号手段と

を備えることを特徴とするデジタル信号処理装置。

18. 上記演奏音信号以外のデジタル信号は、オーディオ信号であることを特徴とする請求の範囲第 17 項記載のデジタル信号処理装置。

19. 上記演奏音信号以外のデジタル信号は、画像信号であることを特徴とする請求の範囲第 17 項記載のデジタル信号処理装置。

20. 上記演奏音信号以外のデジタル信号は、文字を表す信号であることを特徴とする請求の範囲第 17 項記載のデジタル信号処理装置。

21. 上記制御データは、上記演奏音信号以外のデジタル信号の再生タイミングを制御するデータであることを特徴とする請求の範囲第 17 項記載のデジタル信号処理装置。

22. 上記制御データは、上記演奏音信号以外のデジタル信号の

再生に関するパラメータを制御するデータであることを特徴とする請求の範囲第 17 項記載のデジタル信号処理装置。

23. 上記制御データは、上記インターフェースデータの標準フォーマットにおける所定のイベントに、符号化されたデータ列として予め記述されていることを特徴とする請求の範囲第 17 項記載のデジタル信号処理装置。

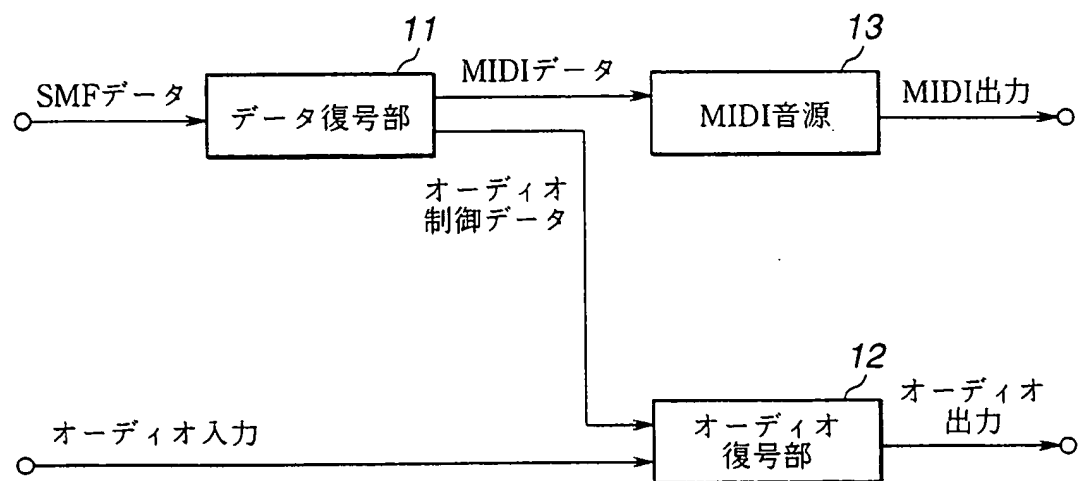
24. 楽器音情報を複数蓄積した音源から出力される演奏音信号に対して、上記演奏音信号以外のデジタル信号を同期させるための制御データを含んだインターフェースデータを生成することを特徴とする制御データ生成方法。

25. 楽器音情報を複数蓄積した音源から出力される演奏音信号に対して、上記演奏音信号以外のデジタル信号を同期させるための制御データを含んだインターフェースデータを生成する手段

を備えてなることを特徴とする制御データ生成装置。

26. 楽器音情報を複数蓄積した音源に楽器の演奏音を発音させるための演奏データを少なくとも含んでいる、楽器演奏のためのインターフェースデータ中に予め符号化されて記述されている制御データを復号するステップと、上記ステップで復号した制御データに基づいて上記演奏データの再生タイミング情報に対応させて上記演奏音の信号以外のデジタル信号を復号するステップとを備えてなるプログラムを記録していることを特徴とするプログラム記録媒体。

1/11

10 デジタル信号処理装置**FIG.1**

FF	イベントの種類	データ長	メタ・イベントデータ
----	---------	------	------------

FIG.2

FF	7F	データ長	メーカID	シーケンス・スペシフィック・イベントデータ
----	----	------	-------	-----------------------

メタ・イベントデータ

FIG.3

FF	7F	データ長	メーカID	制御イベントのID	制御量	制御イベントのID	制御量
----	----	------	-------	-----------	-----	-----------	-----	-------

シーケンス・スペシフィック・イベントデータ
オーディオ制御データ オーディオ制御データ

FIG.4

3/11

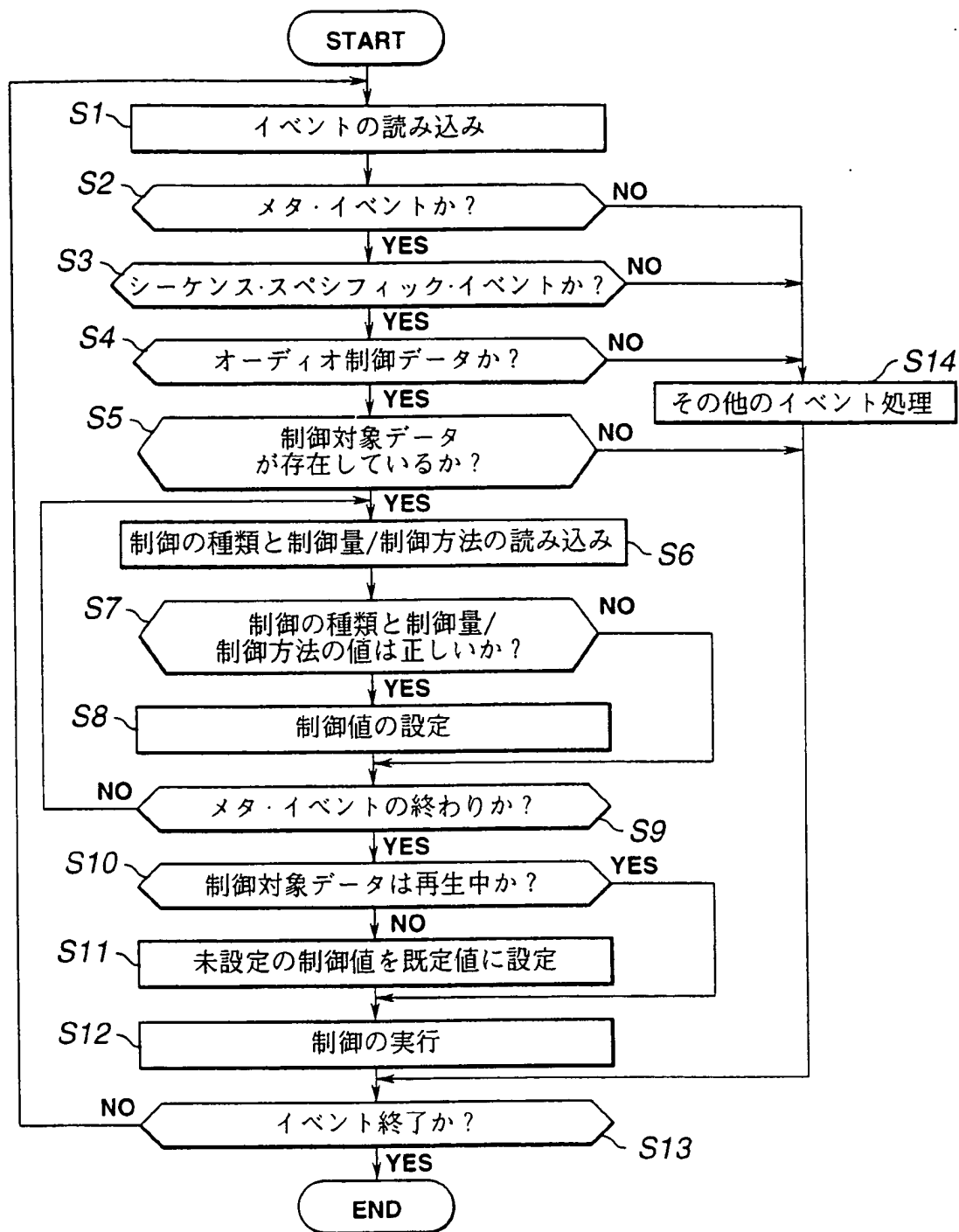


FIG.5

4/11

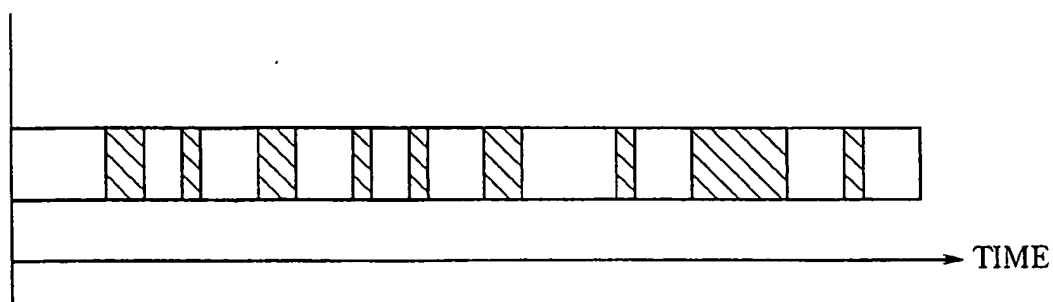


FIG.6A

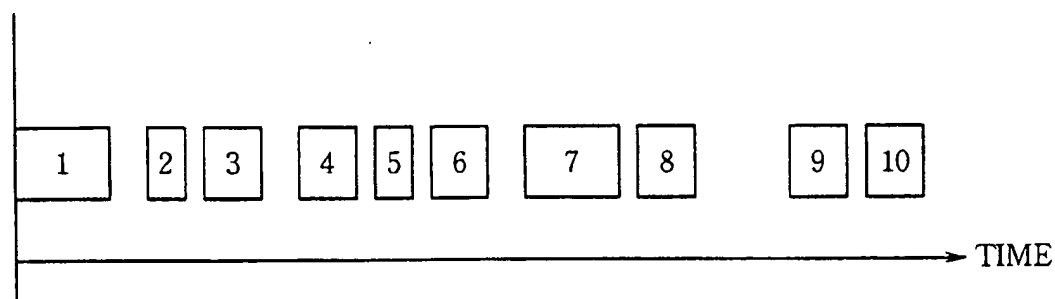


FIG.6B

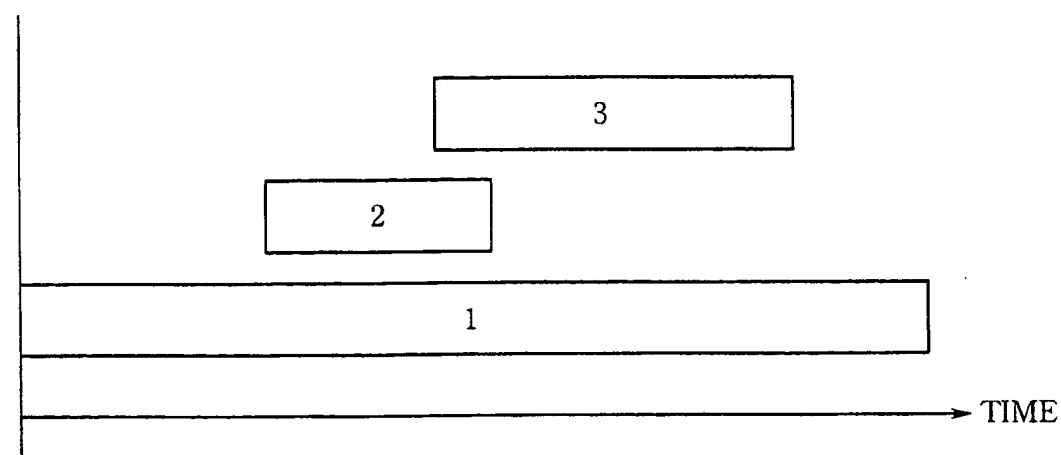


FIG.7

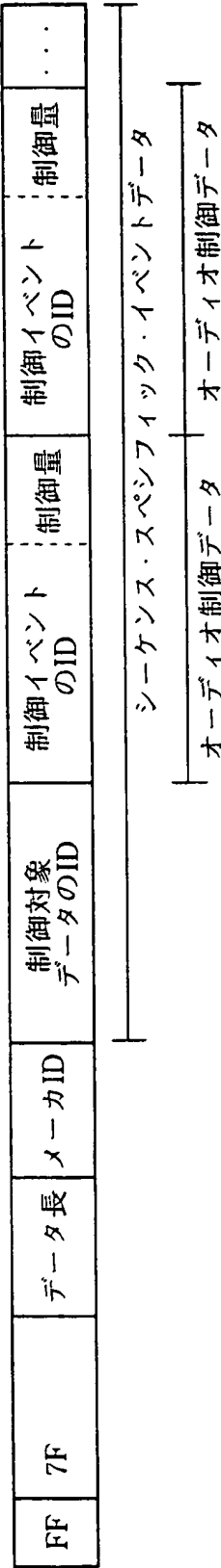


FIG.8

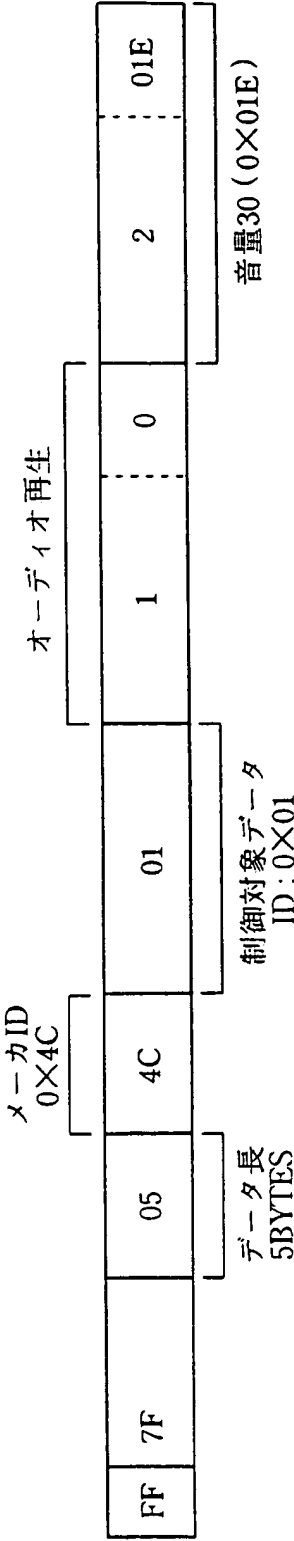


FIG.10

6/11

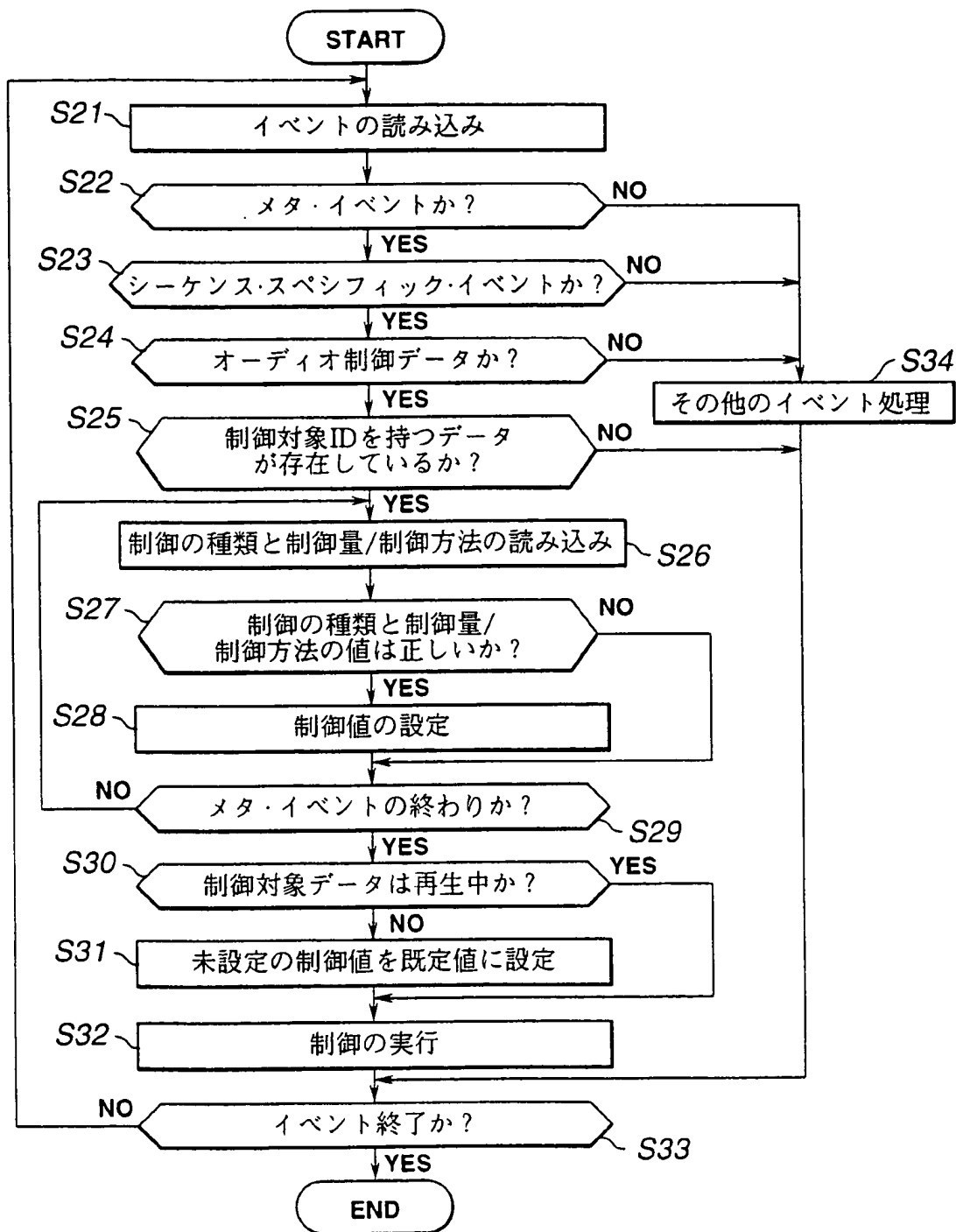


FIG.9

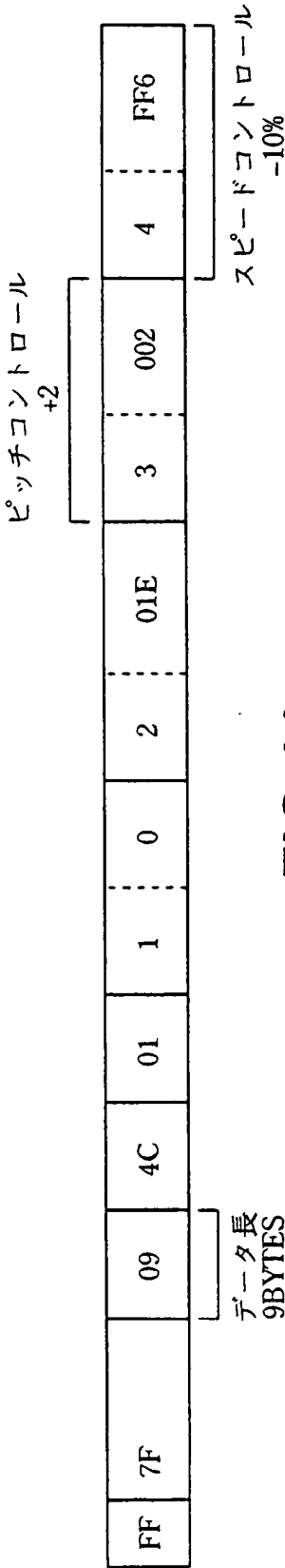


FIG.11

7/11

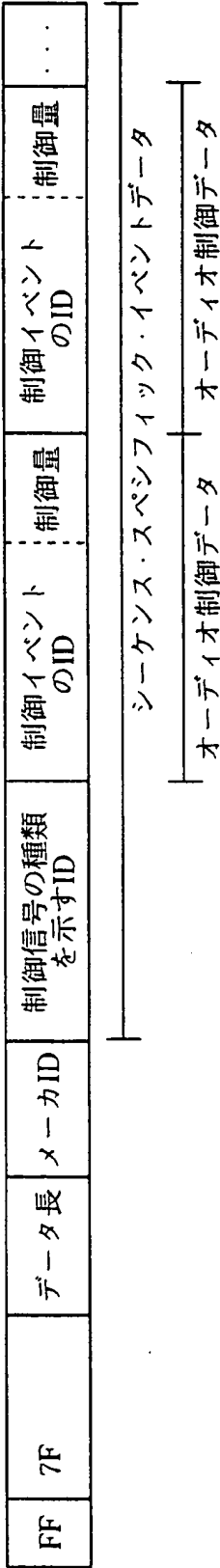


FIG.12